

002d2cd8-0

Simon Austin

COLLABORATORS

	<i>TITLE :</i> 002d2cd8-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Simon Austin	March 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	002d2cd8-0	1
1.1	FGP - Contents	1
1.2	FGP Introduction	1
1.3	FGP Distribution	2
1.4	FGP Instructions	3
1.5	FGP Command Line	3
1.6	FGP Input Files	4
1.7	FGP Old Format Data Files	5
1.8	FGP Data Files	6
1.9	FGP teams.fgp2 File	6
1.10	FGP drivers.fgp2 file	7
1.11	FGP chassis.fgp2 File	8
1.12	FGP engines.fgp2 File	8
1.13	FGP scores.fgp2 File	8
1.14	FGP Output	8
1.15	The error messages	9
1.16	Miscellany	11
1.17	What was used to make FGP	13
1.18	Heddley	13
1.19	EdWordPro	14

Chapter 1

002d2cd8-0

1.1 FGP - Contents

FGP Version 2.10

by~Simon~Austin

Introduction

- What exactly is FGP

Distribution

- What files should be here

Instructions

- How do I use FGP

Miscellany

- Bugs, etc.

1.2 FGP Introduction

What is FGP?

FGP is a program written~primarily~for~people~participating~in~Autosport's~Fantasy Grand Prix~competition.~It~allows the~user to keep~track~of up to~100 individual teams.~This is~useful~as~Autosport only ever~prints~the top~(and bottom <shudder>)~100~teams and~most~people aren't in~there~:-(

Is that all it does?

Pretty much. It will however handle an~entire~season~of~results,~rather~than~the less-than-half season used~in~the~Autosport~competition.

What will it run on?

FGP was primarily developed on the Amiga, but, by only using portable C code, it can also be compiled and run on other platforms. It has successfully been compiled on several Unix systems and I have been informed that it has compiled on OS/2. It won't compile and run under either Windows or MS-DOS (and similar) since it requires reasonable sized filenames.

I've used an earlier version of FGP, what does version 2.10 offer me?

Well first, it's much faster than previous versions. This is not really important in a program like FGP, but the speed-up is quite noticeable, particularly under Unix.

Secondly, it provides an improved version of the standard race output and two new complete season tables detailing the scores obtained by the drivers and the teams in all races so far.

Thirdly, to accomodate disqualified drivers a new form of data file is being used.

Fourthly, it still reads the old data files so you'll only need to create new data files if you want to.

Fifthly, it allows comments in any of the data files it uses (with one exception).

Finally, it provides more error checking of the data files (no more Unknown drivers in the output).

1.3 FGP Distribution

How distributable is FGP?

FGP is freely distributable, although the ©Copyright remains with the Author on the program, the source code and this documentation (whether in plain or formatted ascii, AmigaGuide or HTML format).

Any and all data files included in a distribution archive are placed in the public domain for anyone to use.

The rules for the Autosport Fantasy Grand Prix (the file Rules.doc) are ©Copyright 1994 Haymarket Specialist Motoring Publications Ltd.

Any and all data files created for or during the use of FGP are the property of the user.

What should be in the archive?

The Amiga distribution consists of the files in an LhA archive. The archive will be named fgp???.lha, where ??? indicates the version number. The archive will contain the following files:

FGP.c - the source
 FGP - the actual program
 FGP.doc - this document in ascii format
 FGP.guide - this document in AmigaGuide® format

plus any associated icons. The archive may, depending of the time of distribution, contain one or more data files for use with FGP.

There is no standard non-Amiga distribution yet. Methods of distributing the source and documentation are being looked into.

1.4 FGP Instructions

Instructions for using FGP:

The command line

The input files

The data files

The output

The error messages

Generally the process of producing the output for a particular race is as ↔

follows:

- 1) Create the race data file(s) in any text editor (ed, edit, vim, memacs);
- 2) Run fgp with this data file (fgp data.file);
- 3) Fix any problems (see error messages for more details);
- 4) Re-run fgp to update the scores file (fgp data.file);
- 5) Run fgp to produce the outputs (fgp -r -t -d).

Of course the options in 5 can be combined with the data file in 2 or 4 if you want.

1.5 FGP Command Line

FGP uses a simple command line:

```
FGP <[-r[n]] [-t[n]] [-d[n]] [filename]>
```

Any or all of these options may be given on the command line, in any order, in any case (except for the filename, see below).

-r[n]

Produce the standard single race output, plus a breakdown of each team's score.

-t[n]

Produce a table showing the scores for each team for each race.

-d[n]

Produce a table showing the scores for each driver for each race.

In all three cases, a number after the option indicates the number of races to produce the results for. If no number is given, it is assumed that results for all races are required.

filename

This is either the full filename for the new data file or the root filename (Example for Example.grid, Example.rslt and Example.xtra) for the old data files. The case sensitivity of this option is system-dependant so take care with this.

1.6 FGP Input Files

FGP can use two forms of data file, the new format (detailed here) ←
and

the

old format
(detailed later).

The new format consists of a single ascii text file which lists all the drivers involved in a particular race, their finishing positions, whether they finished, retired, were disqualified or failed to qualify, where they came in the warmup and if they were nominated for the Driver of the Race award.

The format of the file is as follows:

```
## This is a comment, all comments start with ## at the
## start of the line. Comments are optional, you don't have to
## have any, or you can have as many as you want.
Race name
## The Race name uniquely identifies the Race, only the first six
## characters are used so don't call two different races Spongel2 and
## Spongel3 because they won't be identified as different.
Surname of driver at grid position 1
## This is case insensitive, Schumacher is the same as schumacher.
Flags for driver 1
Surname of driver at grid position 2
Flags for driver 2
|
|
```

```

Surname of driver at last grid position
Flags for last driver
Surname of driver who did not qualify
Flags for non qualifier
Surname of second driver who DNQ
Flags

```

The 'Flags' line comprises four items detailing the driver's result. They are:

```
Status Position Warmup Nomination
```

Status is one of FfRrDdNn where F or f means finished, R/r means retired, D/d means disqualified and N/n means did not qualify.

Position is the position the driver finished in. This should be included for retirees and disqualifications as well as those who finished. Drivers who DNQ should be given a finishing position of 0.

Warmup is the position the driver achieved in the Sunday warmup. DNQs generally don't take part in this so they will usually be 0 again. NB there must be six legal drivers here or the program will report an error. If a driver who was in the warmup top six was disqualified the driver who came seventh must be included for the results to be output. Drivers whose positions don't matter can be given either their position or 0.

Nomination is either + for the nominated driver or ~ for those who aren't. Only one nominated driver is allowed and there will be an error message if this is the case.

1.7 FGP Old Format Data Files

The three race data files must be named race.grid, race.rslt and race.xtra where race can be anything, but it must be the same for all three files. Generally I use the name of the host country to make identification easier.

The .grid file details the starting grid of a race and is of the form:

```

no_of_drivers
driver1
driver2
|
|
driverN

```

This file can be created in any text editor (memacs, ed) and contains on its first line the number of drivers in the race including any non-qualifiers. The following N lines are the surnames of the drivers in grid order, non-qualifiers last. Thus for Brazil this file had 28 entries (26 drivers and 2 non-qualifiers), whilst for San Marino there were only 26 (25 drivers and 1 non-qualifier). NB Only the first six letters of each name are actually required, a spelling mistake in these will be reported as an unknown driver by the program.

The `.extra` file details the fastest warmup drivers, the nominated driver, the last classified driver and the last qualifier:

```
warmup1
warmup2
|
warmup6
nominated
last_classified
last_qualifier
```

This file is also a simple text file, the six warmup drivers and the nominated driver are the surnames of the drivers whilst `last_classified` and `last_qualifier` are numbers. The `last_classified` is the position of the last unretired driver, for example in Brazil the number would be 12 since 12 drivers crossed the finish line. NB It is not always the case that you have to cross the line to be classified. In South Africa in 1993 only five drivers made it to the finish line, but the fifth driver was classified seventh since he hadn't completed as many laps as two drivers who crashed before they finished. The `last_qualifier` is normally 26, but this may change if, for various reasons, some teams are not running all their cars.

The `.rslt` file contains the positions of the drivers at the end of the race. All drivers who started the race must be in here, but don't include the non-qualifiers. The drivers must match exactly with those detailed in the `.grid` file otherwise you may get some funny results, drivers appearing twice and suchlike. The file is just:

```
driver1
driver2
|
|
driverN
```

1.8 FGP Data Files

FGP uses five data files to store the driver/team details. These are: ↵

`teams.fgp2`

`drivers.fgp2`

`chassis.fgp2`

`engines.fgp2`

`scores.fgp2`

1.9 FGP teams.fgp2 File

teams.fgp2 details the fantasy teams taking part in the competition. Each team consists of three drivers, a chassis and an engine. Details of how to make a team are in the Rules.

The teams.fgp2 file is constructed as follows:

```
## Comments.
8
## Number of teams in the competition.
## Team 1 details
Owner
Name
Driver 1 Surname
Driver 2 Surname
Test Driver Surname
Chassis
Engine
## Team 2 details
|
|
## Team 8 details
```

1.10 FGP drivers.fgp2 file

drivers.fgp2 is the largest data file used by FGP. It contains the full name and team of each driver in the real Formula One competition. Currently drivers.fgp2 can contain a maximum of 100 drivers and this seems to be sufficient at the moment.

The file is constructed as follows:

```
## Comments can be used, but aren't required.
Surname
First name
Chassis
Engine
```

repeated for each driver. If drivers enter the competition who weren't originally listed, they can be added to the end of the file as required. If a driver moves from one team to another during a season (e.g. Philippe Alliot or Andrea de Cesaris) this should be reflected by changing the chassis and engine names.

The surname and first name should be cased as they will appear in the final output (e.g. de Cesaris, Schumacher), the case of the chassis and engine is irrelevant.

NB: If you change an entry and then re-score an earlier race the change in chassis/engine will be noted with a non-fatal error message. If you did not intend for that driver/car combination to be used for that race then you'll have to change the file to the correct combination and score the race again. This method of warning about changes was implemented in order to allow mistakes to be corrected in prior races in an inobtrusive manner as possible.

1.11 FGP chassis.fgp2 File

chassis.fgp2 is a simple list of all the available~chassis~names as~they~will be used.

Format:

```
Name 1
Name 2
|
Name N
```

1.12 FGP engines.fgp2 File

engines.fgp2 is just a simple list of the available~engine~names as~they~will be used.

Format:

```
Engine 1
Engine 2
|
Engine N
```

1.13 FGP scores.fgp2 File

scores.fgp2 contains the points scored by the drivers in~all~the races~so~far recorded.

This file is not meant to be altered by the~user.~Warning:~altering this~file may result in incorrect results. Note also that this is the only data file in which comments cannot be used.

1.14 FGP Output

FGP can produce three forms of output.~The~output~required~is~indicated~by~the command~line

.

Race~scores

Teams~table

Driver~table

Normally, FGP sends its output to the standard~output.~On~the~Amiga~this~will~result in the output appearing in the~console~from~which~FGP~was~started.~Under Unix the output will appear in

the~shell~you~started~FGP~from.~This~method allows the output to be~redirected~as~required. The~output~may~be~directed into a file or piped~through~another~program (on~the~Amiga,~piping~requires either a second~console or~a~Unix~style~pipe~command~installed).

```
FGP -r >results.txt
FGP >results.txt -r
```

These two commands will produce the standard race~scores~and~save~them~in~a~file called results.txt.

```
FGP -r | more
```

Under Unix (or AmigaDOS with an~appropriate~setup)~this~will~send~the~results to the console/shell but stop~every screen~to~allow~you to~read~the~output.

NB: At least on my Amiga, more tends to~miss~the~first~few~characters~when~reading from a pipe.

```
FGP -r | most
```

Is a better method since Most doesn't miss~the~first~characters~in~the~pipe.~Most is available from the Aminet~in~text/show~directory.

To pipe the output on an Amiga without a~Unix~style~pipe~you'll~have~to~open~a second console. In the first type

```
FGP -r >PIPE:fgp.out
```

In the second console type

```
more <PIPE:fgp.out
```

It's much easier to install a~proper~pipe~though.~The~appropriate~programs~can be found on the~Aminet.~Just~search~for~finkeltools.lha. I~think it's in~util/shell.

1.15 The error messages

When reading in a data file, FGP will produce a message if it finds an error in the file(s). There are two sorts of error message; fatal and non-fatal. Fatal error messages will stop the program immediately, but non-fatal ones won't. If a line number is listed it means the line number in the file not including comments.

Fatal errors are:

```
Usage: FGP <[-r[n]] [-t[n]] [-d[n]] [filename]>
```

This means you didn't give FGP the correct commands.

```
Unable to open "filename".
```

FGP couldn't find one of the files it requires to run. NB if you're using a single input file (e.g. Italy.data), FGP will not report if it is missing but try to find an old format file called Italy.data.grid and report if that is missing

Error in file "filename" on line n. Unknown driver.

FGP encountered, in the named file, a driver's name which it doesn't know. This either means a name has been wrongly spelt or a new driver has driven. In the first case you should correct the spelling, in the second the new driver should be added to the bottom of the drivers.fgp2 file.

Error in file "filename" on line n. Illegal nominated driver.

The nominated driver was not an existing driver.

Error in file "filename" on line n. Illegal final classified driver.

The final classified driver was greater than the actual number of drivers competing.

Error in file "filename" on line n. Illegal last qualified driver.

The last qualified driver was greater than the actual number of drivers competing.

Error in file "filename" on line n. Final classified/last qualified driver mismatch.

The final classified driver was less than the last qualified driver.

Error in file "filename" on line n. Flag not one of FRDN.

The status flag was incorrect.

Error in file "filename" on line n. Illegal finishing position.

The driver's finishing position was not within the range 0 to 26.

Error in file "filename" on line n. Illegal warmup position.

Same error but for the warmup position.

Error in file "filename" on line n. Nominated flag not - or +.

The nominated flag must be - or +.

Error in file "filename" on line n. Too many nominated drivers.

Two drivers have a + as their nominated flag.

Error in file "filename" on line 0. Not enough warmup positions.

At least six drivers must be given a warmup position, if a driver in the top six warmup positions is disqualified, his points are forfeit and the 7th fastest wamup driver should be supplied. NB this error always gives line 0

as the error line.

Non-fatal errors are:

Driver's car has changed from Chassis1 to Chassis2.

Driver's engine has changed from Engine1 to Engine2.

These errors will occur if you re-score a race and FGP detects that when the race was first scored the driver had a different car/engine to this time. If the new combination is correct you don't need to do anything, but if the old combination was correct you'll have to adjust drivers.fgp2 and re-score the race again.

1.16 Miscellany

Bugs:

There may still be some bugs left in FGP. There have been quite a few changes since the last version and, whilst I've tried to fix everything, some things may have slipped through the net.

If you think you've found a bug, please don't hesitate to contact me. You can email me at either of the following addresses:

austins@soll.uel.ac.uk
austins@bkmain.uel.ac.uk

Soll is the preferred address unless you've got a MIME compatible system in which case BKMAIN is better.

If you do report a bug, it would be nice if you could send the data files you were using at the time. Don't worry about which sort of compression or archival to use, I can cope with all the common methods.

The 'impossible Unix bug' section:

There have been a couple of bug reports where a driver's points appeared in the breakdown but not actually in the driver's score. This bug only occurs under Unix and should in fact be impossible. If you do find something like this please contact me as soon as possible and send me the data files you were using.

The 'my output seems funny' section:

There are a couple of things that can happen if there are 'errors' in the data files. If two drivers are given the same finishing position then a strange blank line will appear in the output:

```
7 Berger          9 (00+0+09+0-0-0) etc.
7                  0 (00+0+00+0-0-0) etc.
```

The driver who should be given position 8 has been given an earlier position. A similar error in the warmup positions is less noticeable but the warmup points will be incorrectly allocated.

Another problem can occur if there is a blank line at the end of the drivers.fgp2 file. Normally, blank lines in the data files will be flagged as unknown drivers, but if there is also a blank line in drivers.fgp2 the results may be unusual.

The 'but that's not what Autosport gave them' problem:

With version 2.10 of FGP comes the facility to disqualify drivers. The Autosport rules state that 'Drivers who are disqualified ... will lose all points gained from that weekend.' From the results in Autosport when Michael Schumacher was DQed from the Belgium GP it can be seen that this includes points scored by their cars. Also from the Belgium GP results Damon Hill scored two points by finishing ahead of his grid position. Since he started from third and finished second I assumed that when a driver is DQed all those behind him are considered, for the purposes of the increase on grid position points, to have moved up one position.

However, when Olivier Panis was DQed from the Portuguese GP all those drivers who finished after him were not given this extra point. For example Gianni Morbidelli, who finished in 10th after Panis and started 16th, was only given 6 points even though Autosport listed him as finishing 9th. The program will award 7 points which I believe is correct although I haven't yet contacted Autosport to check this.

Of course, there may be instances where I have really got it wrong, in which case please contact me at the above address.

Version History:

1.00 - First version. Very user-unfriendly, input was by driver/chassis/engine number only. Team results were output in the order the teams were in the input file and drivers were listed by number. No spaces allowed in names.

1.01 - Teams results now output with driver's name. Added Grid/Results output. Due to a mind block placed on me by aliens I was unable to output the grid/results as names and had to just output the driver's number. The results only listed up to the last classified driver.

1.02 - Mind block removed. Grid/results now displayed with names and including retired drivers. Tidied up the team results to use more white space.

1.03 - Fixed no spaces problem.

2.00 - Total rewrite. Allowed data input as drivers name, used separate data file to hold drivers chassis/engine. New output format suggested by J Simpson.

2.01 - Added facility to handle races with other than 28 entrants (26 qualifiers and 2 non-qualifiers). Added -/+u command line to stop FGP

updating the scores file. Kludged ``bug`` causing confusion between Ferrari chassis and Ferrari engine. Fixed bug causing confusion between Ford Zetec-R and Ford HB V8.

2.02 - Increased length of names from 20 characters to 80, increased number of possible teams from 10 to 50. General tidying up of code. Changed output slightly.

2.02a - Uses new rules

2.02b - Further rule changes

2.03 - Bug fixed: If a team had a negative score it would be replaced by team 0 (which consisted mainly of Unknowns). Some code tidying and a potential bug fixed.

2.10 - Added: New output, new input file. Speeded up program. Comments allowed in data files. Enhanced error checking. Entire seasons's results recorded

1.17 What was used to make FGP

FGP was created on an Amiga 1200 with 4MB and a 163MB hard drive.

The AmigaGuide® and HTML docs were created using a combination of
Heddley
and
EdWordPro
.

Heddley
is ©1994 Edd Dumbill.
EdWordPro
is ©1994 Martin Reddy.

Many thanks must go to John Simpson for help with various parts of the program. Thanks also to Alex & Lynne, Dave, Chris, David, Martin and TIM for their help in testing FGP.

1.18 Heddley

NAME

Heddley

SYNOPSIS

Heddley FILE, FONT/K, FONTSIZE/K/N, TEXTWIDTH/K/N, EDITOR/K,
FGPEN/K/N, BGPEN/K/N, HELPFILE/K

DESCRIPTION

Heddley is a hypertext editor for AmigaGuide®. It is capable of reading files in AmigaGuide® format and outputting them in

AmigaGuide®, HTML and ASCII with ANSI control codes. Heddley allows you to edit documents, create buttons and apply styles to the document texts.

USAGE

Heddley can be started from the CLI or from Workbench. The command line arguments are the same as the tooltypes:

```
FILE          The AmigaGuide® file to edit.
FONT          The font to use in the edit window (fixed-width).
              Default topaz.font
FONTSIZE     Point size of the edit window font. Default 8.
TEXTWIDTH    Justification width for text. Default 78.
EDITOR       The name of your favourite text editor.
FGPEN       The number of the pen to use for Text pen.
BGPEN       The number of the pen to use for Back pen.
HELPPFILE   The path of the file "Heddley.guide"
```

BUGS

Probably quite a few, but none known at this minute.

SEE ALSO

"Heddley.guide", by Edd Dumbill.

AUTHOR

Edd Dumbill (ejad-a@minster.york.ac.uk)

1.19 EdWordPro

=====

----- Profession V4.1 -----

GENERAL INTRODUCTORY TEXT

MARTIN REDDY, 1994

INTRODUCTION

~~~~~

EdWord is an extremely powerful, general purpose text editor which offers a host of useful and helpful facilities. It was initially released as a

Shareware program, but since that time it has undergone a myriad of improvements and updates resulting in the current version which provides an indispensable aid for every Amiga user. As you might expect, the basic function of EdWord is to edit text and in this role, it offers all the rudimentary facilities which any text editor might offer, but what makes EdWord that bit more special are the extra options which it provides. The following is a list of some of the more notable functions which are supported by version 4.1 :-

- Multiple Documents: you can edit up to 15 documents at once (memory permitting, of course) with a split screen option to view two of them together.
  - A Macro Facility: lets you record any sequence of keypresses/ mouse clicks and menu selections, and then play them back. This is very useful for automating small repetitive jobs.
  - An ARexx port offering an extensive and comprehensive range of commands to provide full automation of EdWord; with the ability to enter these commands directly (without using ARexx), as well as run any ARexx script from within the editor.
  - PowerPacker Support: EdWord will (optionally) decrunch any data files which have been packed with this cruncher
  - A new feature called Auto Suggest whereby if you type in a filename which doesn't exist when loading, then EdWord will make an intelligent suggestion as to what you actually meant to type in!
  - AppWindow Support: When running EdWord on a WorkBench screen, it is possible to load a file by simply dragging its icon into the editor's window
  - A calculator which lets you perform basic arithmetic as well as convert numbers between different bases including decimal,hexa- decimal,octal and binary
  - An auto indenting facility as well as a symbolic indenting facility (i.e. EdWord can automatically indent your source code as you enter it)
  - Text Casing of language keywords. i.e. certain keywords can be forced into a certain case (Uppercase,Lowercase or Capitalised) Useful to maintain a consistent look to a piece of source code
  - Access to AmigaDOS to allow the user the ability to run any DOS command (This could be used to run a compiler or assembler through the editor)
  - The ability to use either the standard EdWord file requester or the Arp or Asl library ones
  - An ASCII table of printable character codes with the ability to insert any such character into the current text (from ASCII code 0 to 255)
  - The ability to sort a block alphabetically - this could be used for arranging a list of names, games etc.
  - checks RAM for any viruses or suspicious programs when it first loads up
-

and alerts the user if it finds anything strange.

- Various screen resolutions are supported including PAL and NTSC standards, custom or WorkBench screen and the ability to work in interlace mode.
  - Powerful printing options including the setting of margins, the page length, the ability to include page numbers and to specify the print pitch and quality.
-